Data Mining

Introduction to Business Intelligence, Data Warehouse and DFM

Claudio Sartori

Department of Computer Science and Engineering – University of Bologna, Italy

Academic Year 2020/21

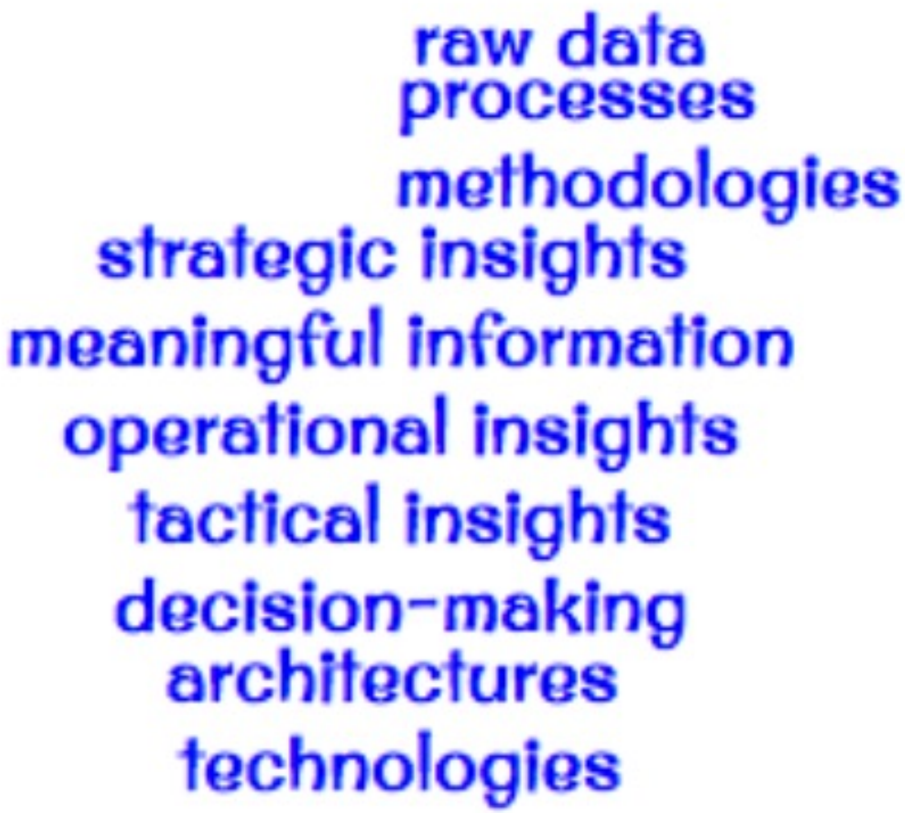
Credits: Federico Ravaldi and Elisabetta Turicchia (Iconsulting)

Summary

1. Main concepts on Business Intelligence and Data Warehousing.
2. Online Analytical Processing (OLAP).
3. Extraction, Transformation and Loading (ETL).
4. Data Warehouse architectures.
5. Conceptual modeling: the Dimensional Fact Model (DFM).

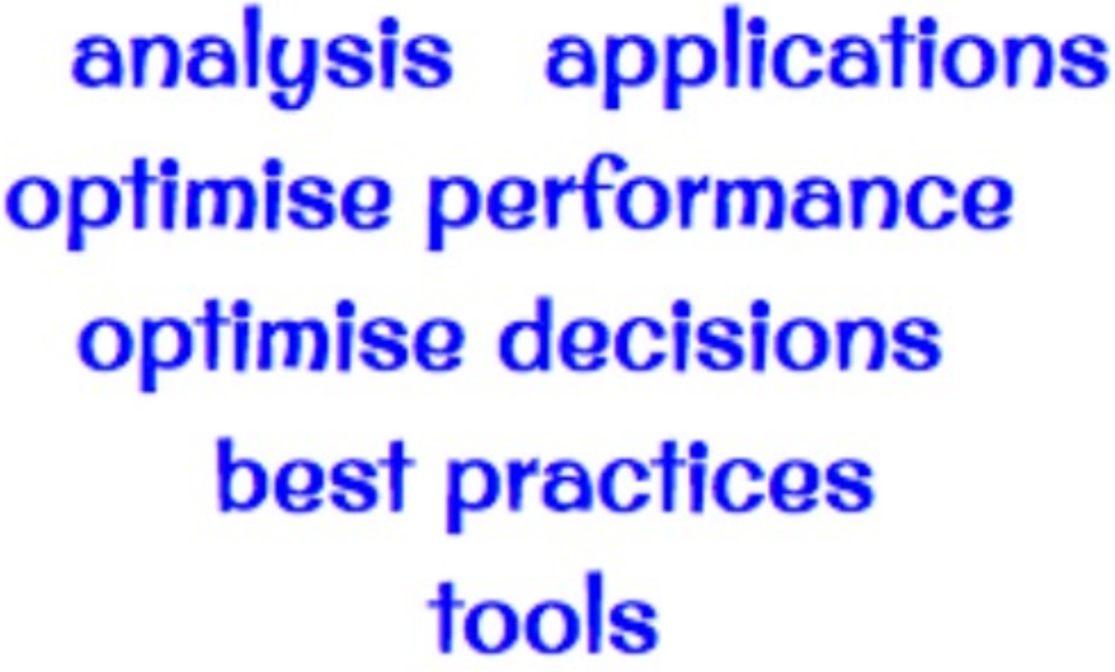
Introduction to Business Intelligence

Business Intelligence – Forrester Research



raw data
processes
methodologies
strategic insights
meaningful information
operational insights
tactical insights
decision-making
architectures
technologies

Business Intelligence – Gartner



analysis applications

optimise performance

optimise decisions

best practices

tools

Business Intelligence - Definitions

Business intelligence (BI) is an umbrella term that includes the applications, infrastructure and tools, and best practices that enable access to and analysis of information to improve and optimize decisions and performance. **Gartner**

Business Intelligence is a set of methodologies, processes, architectures, and technologies that transform raw data into meaningful and useful information used to enable more effective strategic, tactical, and operational insights and decision-making. **Forrester Research**.

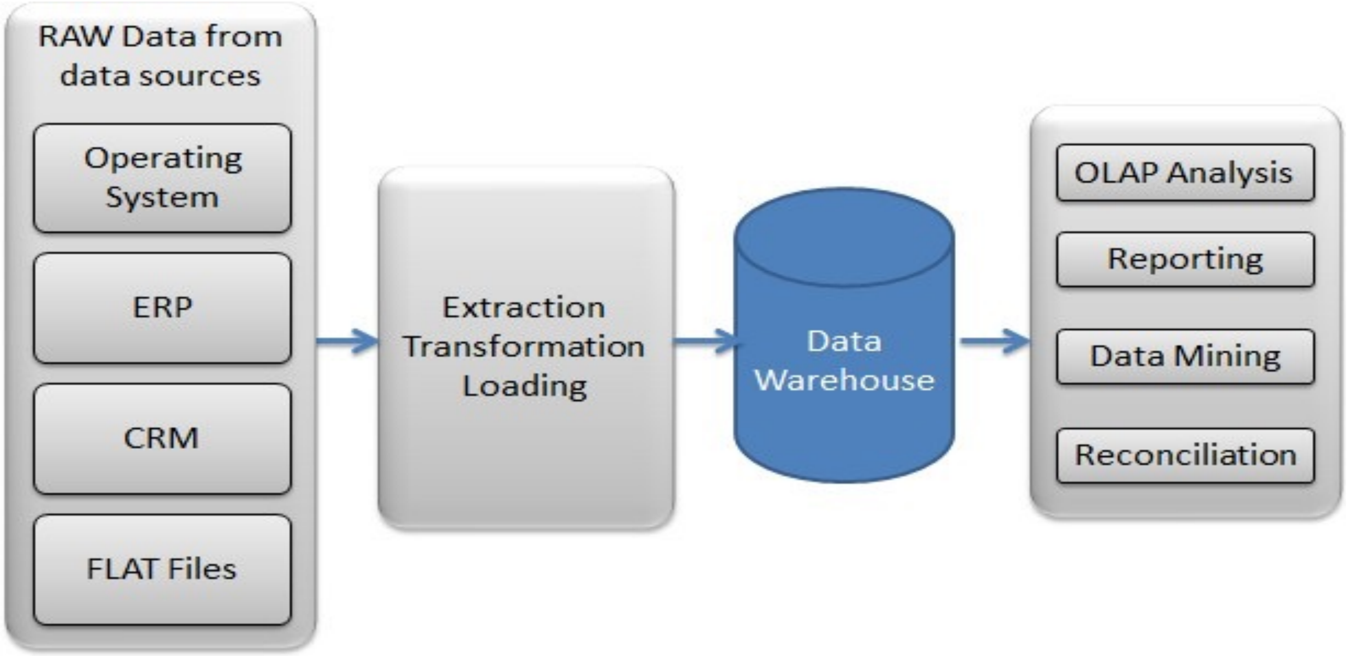
Business Intelligence – Our definition

Process of

- transforming raw data into useful information to support effective and aware business strategies
- capturing the business data and getting the right information to the right **people**, at the right **time**, through the right **channel**

BI and Data Warehouse

The Data Warehouse (DWH) is one of the **main tool to support BI**.



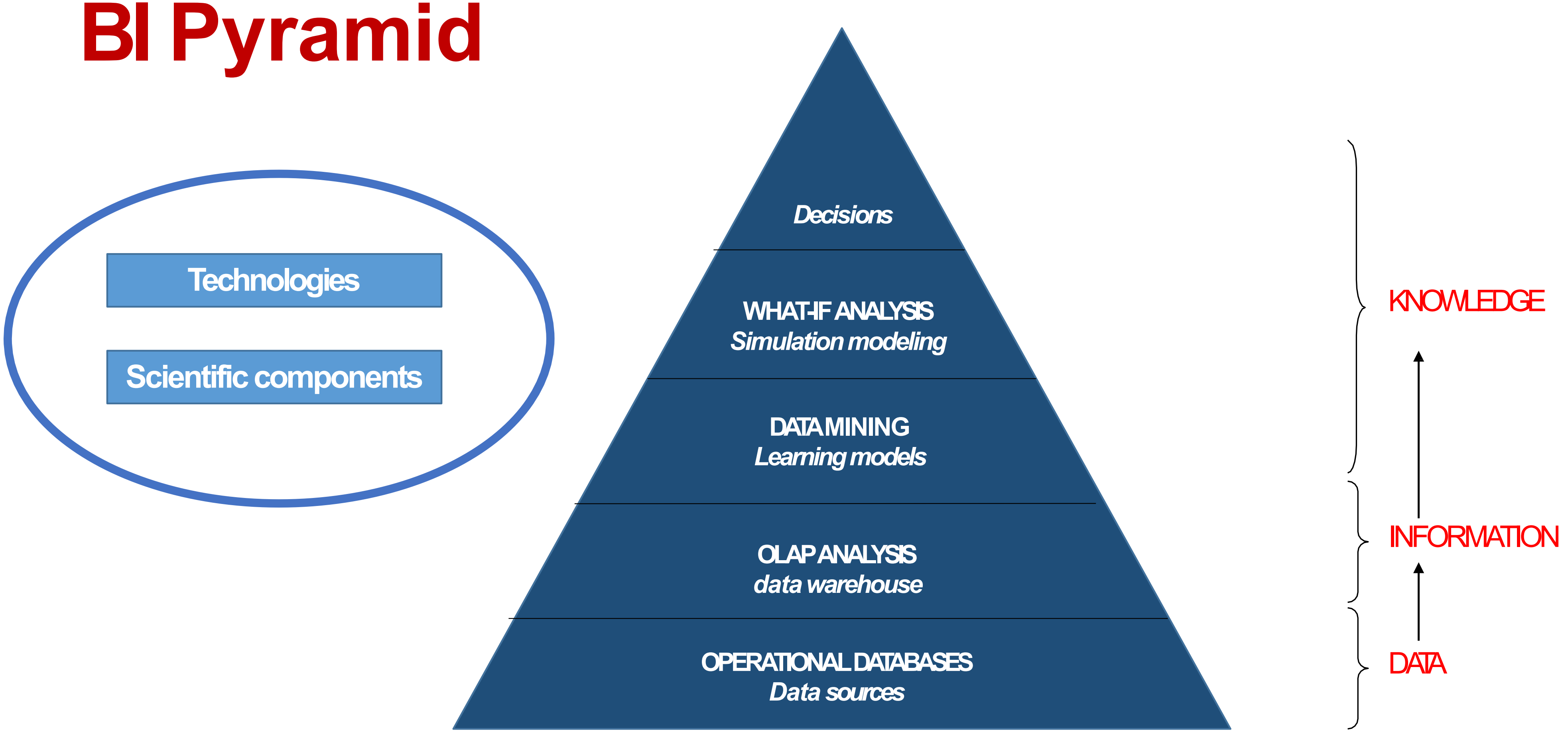
Source layer **Data Staging** **Data Warehouse** **Front-end/Applications**

BI Platform

An ad-hoc infrastructure (both hardware and software) is necessary to allow flexible and effective business analysis:

- Ad-hoc Hardware
- Network infrastructure
- Databases
- Data Warehouse
- Front-end software (Data Visualization)

BI Pyramid


Technologies
Scientific components
Decisions
WHAT-IF ANALYSIS
Simulation modeling
DATA MINING
Learning models
OLAP ANALYSIS
data warehouse
OPERATIONAL DATABASES
Data sources
KNOWLEDGE
INFORMATION
DATA

Main concepts on Data Warehousing

BI and Data Warehouse

- The Data Warehouse (DWH) is one of the main tool to support BI.
- Informally, a DWH is an optimized repository that stores information for the decision-making process.
- DWs are a specific type of Decision Support Systems (DSSs).

As a matter of fact, the increasing number of information a company has to take into account to find relevant business strategies implies more sophisticated solutions than operational databases.

Advantages of DW systems

- They provide the ability to manage sets of historical data
- They provide the ability to perform multidimensional analyses accurately and rapidly
- They are based on a simple model that can be easily learned by its users
- They are the basis for indicator-calculating systems.

Data Warehouse (DWH)

A Data Warehouse is a collection of data that supports decision-making processes. It provides the following features:

- It is **subject-oriented**: it focuses on enterprise-specific concepts, such as customers, products, sales etc.
- It is **integrated and consistent**: a DWH integrates data from different and heterogeneous sources and should provide a unified view of all the data.
- It shows **its evolution over time** and it is **not volatile**: the changes to the data in the database are tracked and recorded so that reports can be produced showing changes over time. Data in the data warehouse is never over-written or deleted -- once committed, the data is static, read-only, and retained for future reporting.

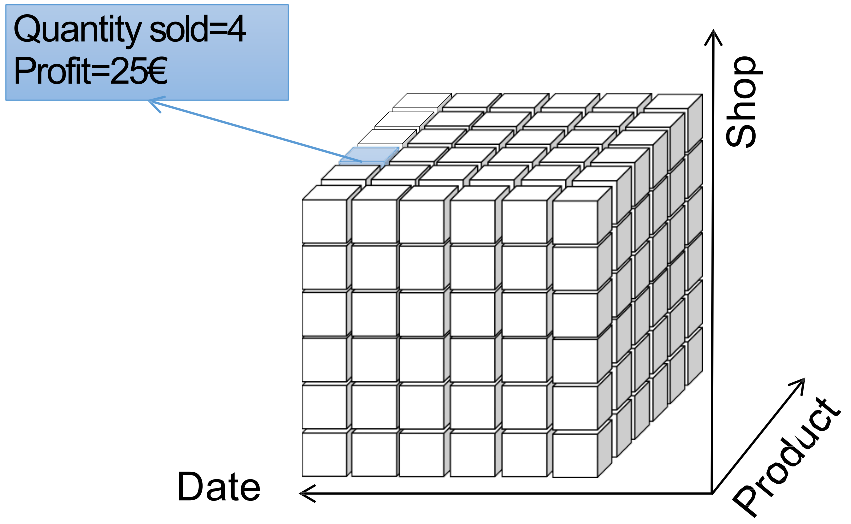
DWH – Examples of fields of application

- **Trade**: sales and claims analyses, shipment and inventory control, customer care and public relations.

- **Financial services**: risk analysis and credit cards, fraud detection.
- **Transport industry**: vehicle management.
- **Telecommunication services**: customer profile analysis, network performance analysis.

- **Health care service**: patient admission and discharge analysis.

Multidimensional Model



On-Line Analytical Processing (OLAP) allows users to interactively navigate the data warehouse information exploiting the multidimensional model. Typically, the data are analyzed at different levels of aggregation, by applying subsequent OLAP operators, each yielding one or more different queries.

Product → Sub-category → Category

Examples of OLAP queries

- Which products maximize the profit?
- What is the total revenue per product category andstate?
- What is the relationship between profits gained by two different products?
- What is therevenue trend in the last three years?

OLTP and OLAP

Online Transactional Processing (OLTP):

- An interactive data processing system based on transactions.
- Each transaction reads and writes a few number of records from tables characterized by simple relationships.
- OLTP systems have an essential workload core “frozen” in application programs.

Online Analytical Processing (OLAP):

- An interactive data processing system for dynamic multidimensional analyses.
- Each query involves huge amount of records to process a set of numeric data summing up the performance of an enterprise.
- The workload changes over time.

Database vs Data Warehouse

Features	Operational Databases	Data Warehouses
Users	Thousands	Hundreds
Workload	Preset transactions	Specific analysis queries
Access	To hundreds of records, write and read mode	To millions of records, mainly read-only mode
Goal	Depends on applications	Decision-making support
Data	Detailed, both numeric and alphanumeric	Summed up, mainly numeric
Data Integration	Application-based	Subject-based
Quality	In terms of integrity	In terms of consistency
Time coverage	Current data only	Current and historical data
Updates	Continuous	Periodical
Model	Normalized	Denormalized, multidimensional
Optimizations	For OLTP access to a database part	For OALP access to most of the database

Data Mart

A Data Mart is a subset or an aggregation of the data stored to a primary data warehouse. It includes a set of information pieces relevant to a specific business area, corporate department, or category of users.

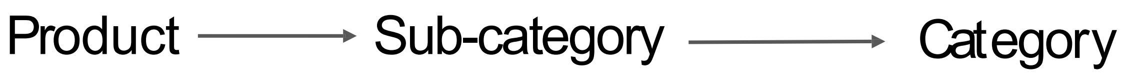
- DMs are used as building blocks while incrementally developing DWHs.
- DMs mark out the information required by a specific group of users to solve queries.
- DMs can deliver better performance because they are smaller than primary DWHs.

Online Analytical Processing (OLAP)

OLAP

OLAP analyses allow users to interactively navigate the DWH information. Typically, the data are analysed at different levels of aggregation, by applying subsequent OLAP operators, each yielding one or more different queries.

OLAP Session: the user can scout the multidimensional model choosing the next operator based on the outcome of the previous ones. In this way, the user creates a navigation path that corresponds to an analysis process for facts according to different points and at different detail levels.



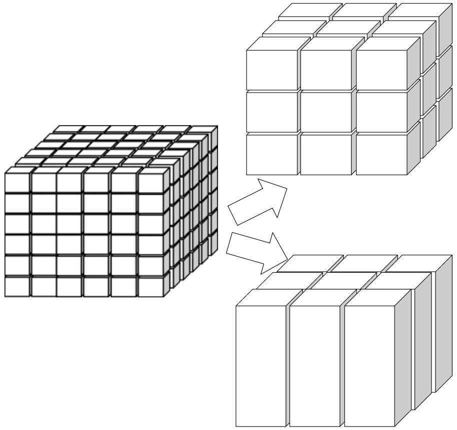
Product → Sub-category → Category

OLAP Operators

- Roll-up
- Drill-down
- Slice-and-dice
- Pivot
- Drill-across
- Drill-through

Roll-up

Roll-up: causes an increase in data aggregation and removes a detail level from a hierarchy.

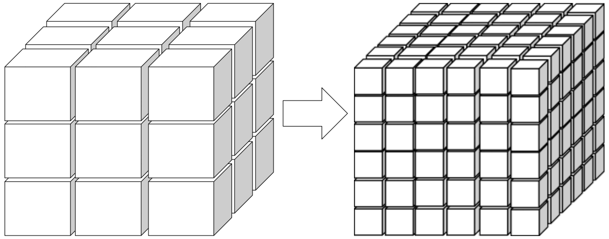


			2015		2014	
			Jan-15	Feb-15	Jan-14	Feb-14
Category	**Type**	**Product**				
Food and Beverages	Dairy products	White milk	90	90	60	80
		Chocolate milk	60	80	70	70
		Yogurt XY	20	30	30	35
	Beverages	Cola	20	10	35	30
		Orange Juice X	50	60	60	45

	2015		2014	
	Jan-15	Feb-15	Jan-14	Feb-14
Type				
Dairy products	170	200	160	185
Beverages	70	70	95	75

Drill-down

Drill-down: is the complement to the roll-up operator; it reduces data aggregation and adds a new detail level to a hierarchy (e.g., from category to subcategory).

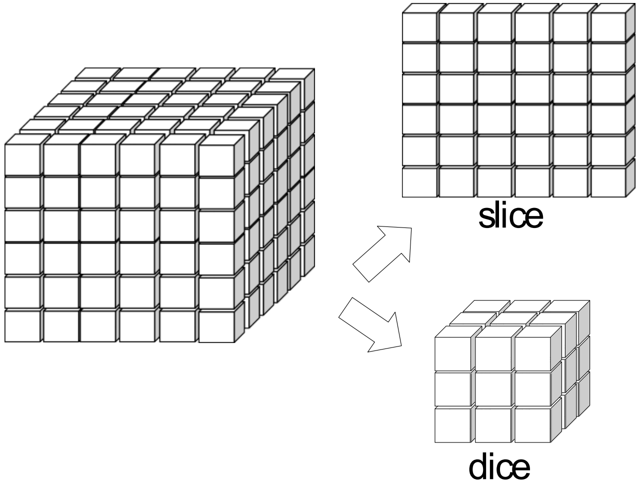


	2015	2014
Type		
Dairy products	370	345
Beverages	140	170

			2015		2014	
			Jan-15	**Feb-15**	**Jan-14**	**Feb-14**
Category	**Type**	**Product**				
Food and Beverages	Dairy products	White milk	90	90	60	80
		Chocolate milk	60	80	70	70
		Yogurt X	20	30	30	35
	Beverages	Cola	20	10	35	30
		Orange Juice X	50	60	60	45

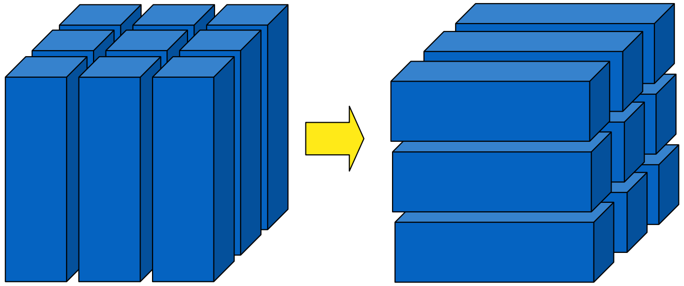
Slide-and-dice

Slice-and-dice: reduces the number of cube dimensions after setting one of the dimensions to a specific value (e.g., category=‘Food and Beverages'); the dicing operation reduces the set of data being analysed by a selection criterion



Pivot

Pivot implies a change in layouts, aiming at analysing a group of data from a different viewpoint.

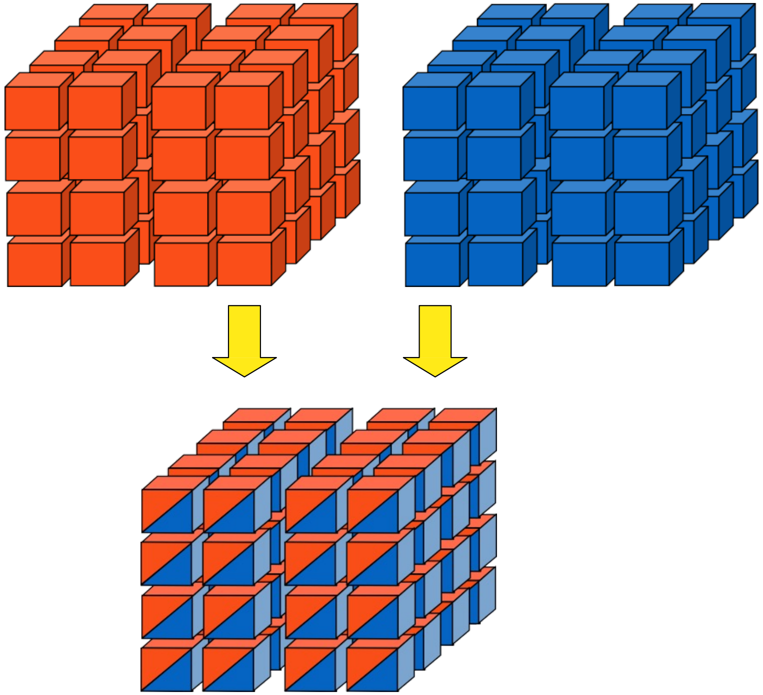


	2015	2014
Type		
Dairy products	370	345
Beverages	140	170

Type	Year	Quantity sold
Dairy products	2015	370
Dairy products	2014	345
Beverages	2015	140
Beverages	2014	170

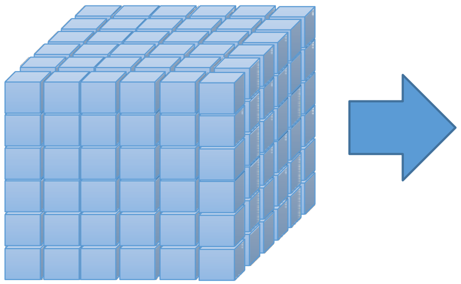
Drill-across

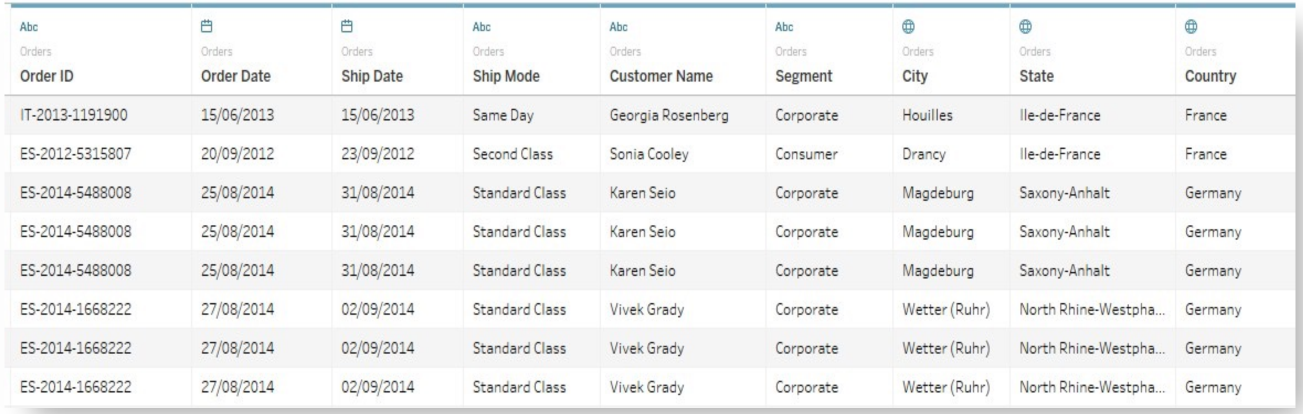
Drill-across allows to create a link between concepts in interrelated cubes,to compare them.



Drill-through

Drill-through switches from multidimensional aggregate data to operational data in sources or in the reconciled layer.



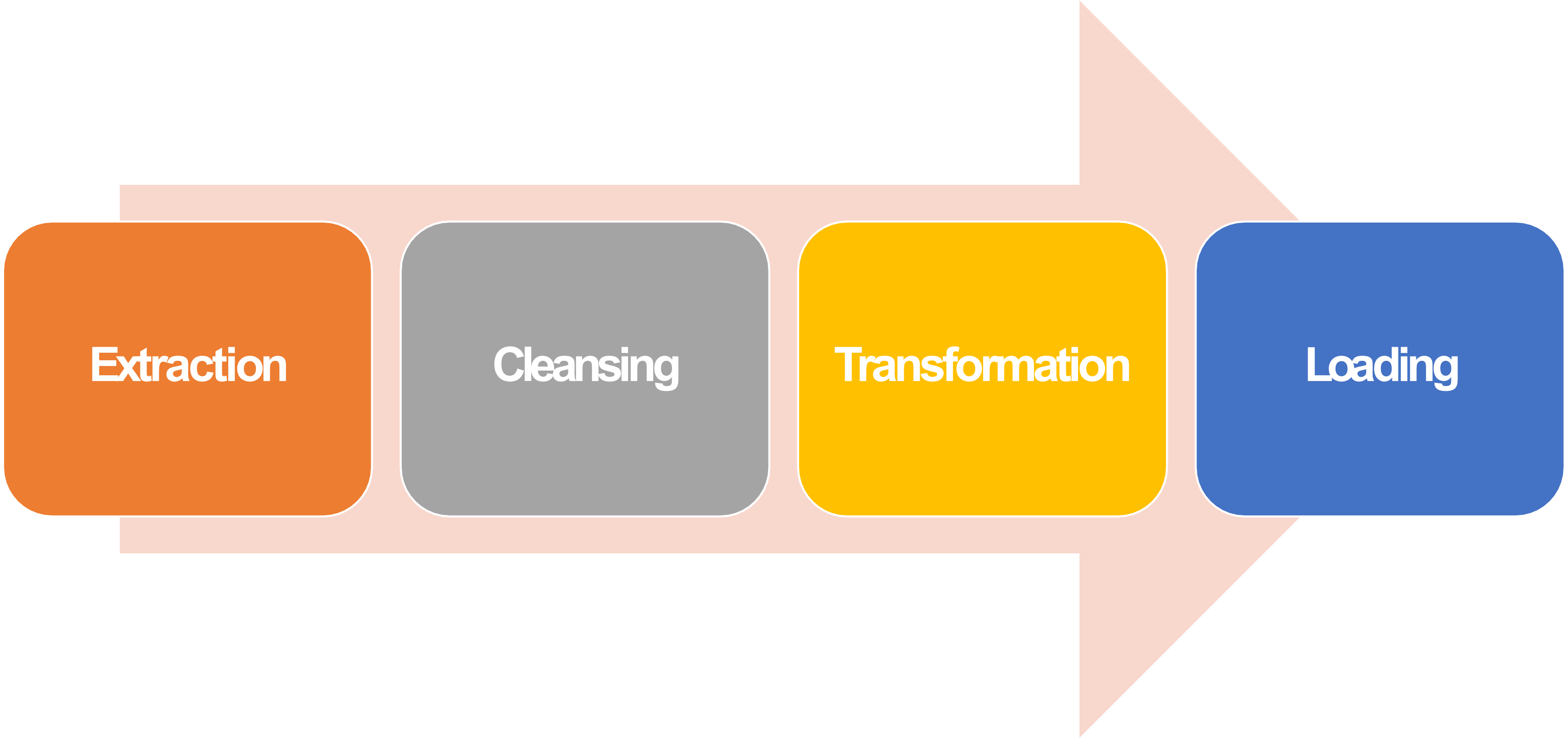


Order ID	Order Date	Ship Date	Ship Mode	Customer Name	Segment	City	State	Country
IT-2013-1191900	15/06/2013	15/06/2013	Same Day	Georgia Rosenberg	Corporate	Houilles	Ile-de-France	France
ES-2012-5315807	20/09/2012	23/09/2012	Second Class	Sonia Cooley	Consumer	Drancy	Ile-de-France	France
ES-2014-5488008	25/08/2014	31/08/2014	Standard Class	Karen Seio	Corporate	Magdeburg	Saxony-Anhalt	Germany
ES-2014-5488008	25/08/2014	31/08/2014	Standard Class	Karen Seio	Corporate	Magdeburg	Saxony-Anhalt	Germany
ES-2014-5488008	25/08/2014	31/08/2014	Standard Class	Karen Seio	Corporate	Magdeburg	Saxony-Anhalt	Germany
ES-2014-1668222	27/08/2014	02/09/2014	Standard Class	Vivek Grady	Corporate	Wetter (Ruhr)	North Rhine-Westpha...	Germany
ES-2014-1668222	27/08/2014	02/09/2014	Standard Class	Vivek Grady	Corporate	Wetter (Ruhr)	North Rhine-Westpha...	Germany
ES-2014-1668222	27/08/2014	02/09/2014	Standard Class	Vivek Grady	Corporate	Wetter (Ruhr)	North Rhine-Westpha...	Germany

Extraction, Transformation and Loading (ETL)

ETL

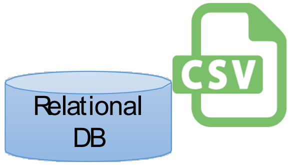
The ETLprocess extracts, integrates, and cleans data from operational sources to feed the Data Warehouse layer.



Extraction

This phase includes the extraction of information from sources.

Types of data



Structured data

Unstructured data

Information with no pre-defined data model

Types of extraction

Static: a DWH is populated for the first time. It is a snapshot of operational data.

Incremental: it is used to update the DWH regularly. It includes the changes applied to source data since the latest extraction. It is based on:

- Timestamp associated to operational data
- Triggers associated with change transactions for relevant data

Cleansing

Procedures to improve the quality of data: standardize data and correct mistakes and inconsistencies.

- **Duplicate data**: for example, a customer is recorded many times in the customer database due to multiple registrations in different shops.
- **Missing data**: such as the customer's age.
- **Unexpected use of fields**: such as a note field used improperly to store the phone as number.
- **Impossible or wrong values**: such as 30th Feb 2016.
- **Inconsistent values for a single entity because different practices were used**: such as University of Bologna rather than Univ. of Bologna.
- **Inconsistent values for own individual entity because of typing mistakes**: such as Oxford Steet instead of Oxford Street.

Solutions for data inconsistencies

Each type of problem requires different techniques for its solution. We can distinguish three main techniques:

- **Dictionary-based techniques**: they are used to check the correctness of the attribute values based on *lookup tables* and *dictionaries* to search for abbreviations and synonyms. We can apply these techniques if the domain is known and limited. These techniques are suitable for solving problems such as typing mistakes and format discrepancies.
- **Approximate merging**: we use this technique when we need to merge data coming from different sources and we don't have a common key to identify matching tuples
    - *Approximate join*
    - *Similary functions*
- **Ad-hoc algorithms**: custom algorithms based on specific business rules (e.g. financial context the following rule must always be checked, *profit=receipts-expenses*)

Dictionary-based techniques (1)

To solve problems related to format discrepancies

Source A

Includes abbreviations
of states

State	..	..
IT		
FR		
DE		
..		

Source B

Includes long descriptions
for the state attribute

State	..	..
Spain		
Italy		
France		
..		

Lookup-table

State Short Desc.	State Long Desc.
IT	Italy
FR	France
DE	Germany
GR	Greece
ES	Spain
..	

Dictionary-based techniques (2)

To solve inconsistencies between correlated attributes.

Source A

Customer ID	Customer city	Customer province
C00001	Bologna	BO
C00002	Cesena	FC
C00003	Cesena	CE
..		

Wrong association

Lookup-table

City	Province
Bologna	BO
Imola	BO
Cesena	FC
Ferrara	FE
..	..

Correct association

Approximate join

In this example we don’t have a common key to join the information because *Customer Code* <> *Customer ID*. The join will be performed on the basis of the common attributes *Customer Address and Customer Surname*. These attributes are not identifier for the customer, thus they are not subject to control procedures to ensure integrity constraints and absence of entry errors. Under those conditions, we talk about *approximate join*.

Marketing Database

Customer
Customer Code
Customer Address
Customer Name
Customer Surname
….

Orders Database

Orders
Order ID
Customer ID
Customer Surname
Customer Address
….

Approximate join on **Customer Address** and **Customer Surname**

Similarity approach

We use the similarity approach to identify different instances of the same information (e.g. a customer has been entered into the same database more times due to typo mistakes).

Customer ID	Customer name	Customer surname
C00001	Elisa	Turricchia
C00002	Elisa	Turicchia
C00003	Mario	Rossi
..		

These two rows refer to the same customer

We can use **affinity functions** (e.g. Edit Distance) to calculate the similarity between two words (in this case the values of the customer surname). If the similarity is higher/lower (it depends on the affinity function we used) than a specific threshold, the two words are the same and we can merge the rows.

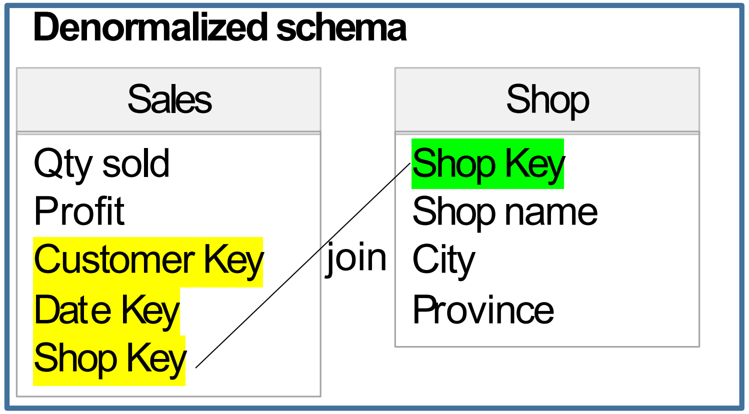
Transformation

In this phase, data from sources is properly transformed to adjust its format to the reconciled schema.

Categories of transformation	Examples
Conversion: changes on data types and format	• Date conversion: from date to number (12/11/2018 →20181112) • String conversion: lowercase to uppercase (unibo→UNIBO) • Naming convention transformation: short description to long description (IT→Italy)
Enrichment: combination of one or more attribute to create new information.	• Calculation of derived data Profit = Receipts - Expenses
Separation/Concatenation	• Attributes concatenation (e.g. customer surname || customer name) • Denormalization/Normalization process. Typically, in the DWH the data is denormalized.

Denormalization

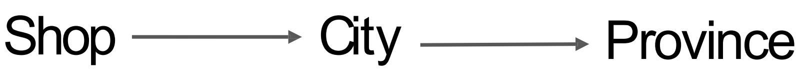
On relational database, denormalization is a breach in third normal form, thus we have table redundancy to reduce the number of joins that we have to perform and speed up the query process.

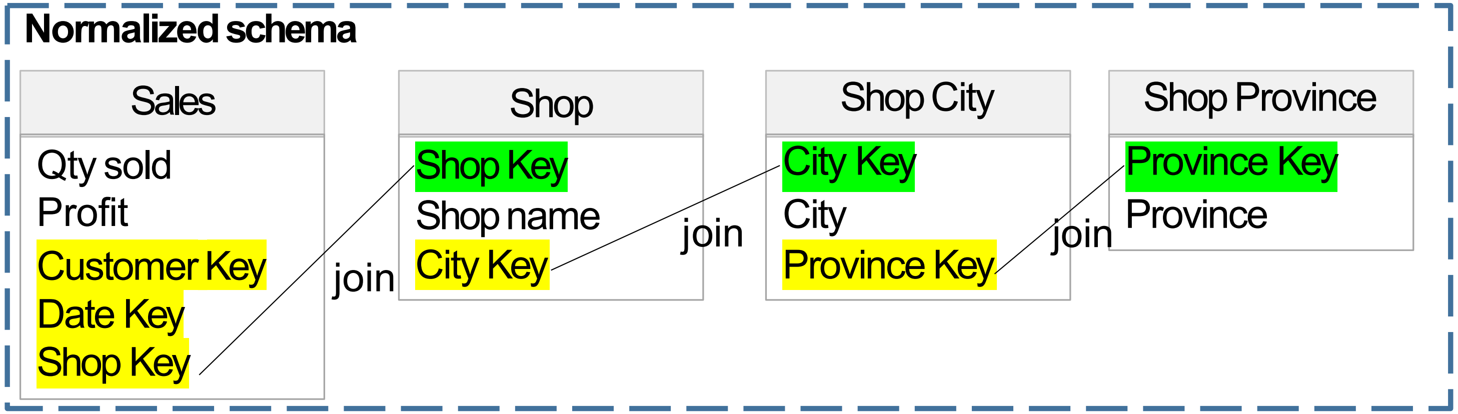


Denormalized schema

Sales
Qty sold
Profit
Customer Key
Date Key
Shop Key

Shop
Shop Key
Shop name
City
Province





Normalized schema

Sales
Qty sold
Profit
Customer Key
Date Key
Shop Key

Shop
Shop Key
Shop name
City Key

Shop City
City Key
City
Province Key

Shop Province
Province Key
Province

Loading

Loading data into a DWH. Two different ways:

- **Refresh**: the DWH is completely rewritten (i.e. older data is replaced). It's used in combination with static extraction.
- **Update**: only those changes applied to source data are added to the DWH. Preexisting data is not deleted or modified. It's used in combination with incremental extraction to regularly update the DWH.

Data Warehouse Architectures

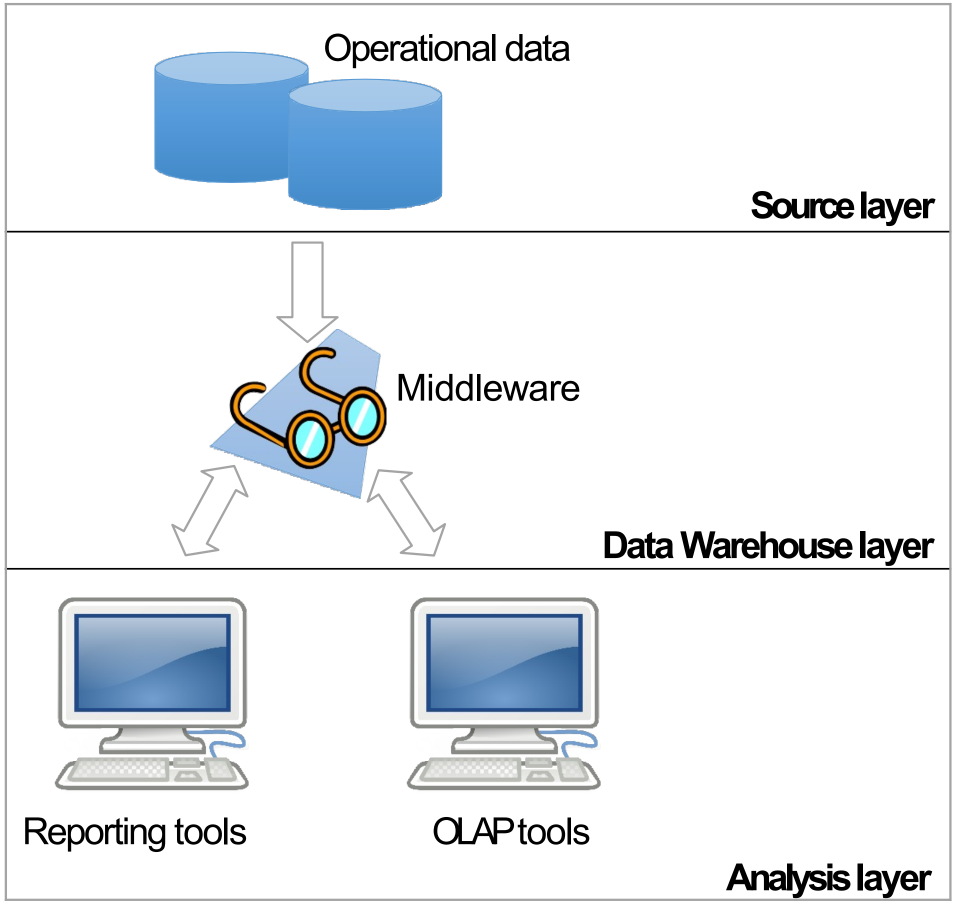
Architectures - Requirements

- **Separation**: analytical and transactional processing should be kept apart as much as possible.
- **Scalability**: hardware and software architectures should be easy to upgrade as the data volume, which has to be managed and processed, and the number of users' requirements, which have to be met, progressively increase.
- **Extensibility**: the architecture should be able to host new applications and technologies without redesigning the whole system.
- **Security**: monitoring accesses is essential because of the strategic data stored in data warehouses.
- **Administrability**: DWH management should not be overly difficult.

Single-Layer Architecture

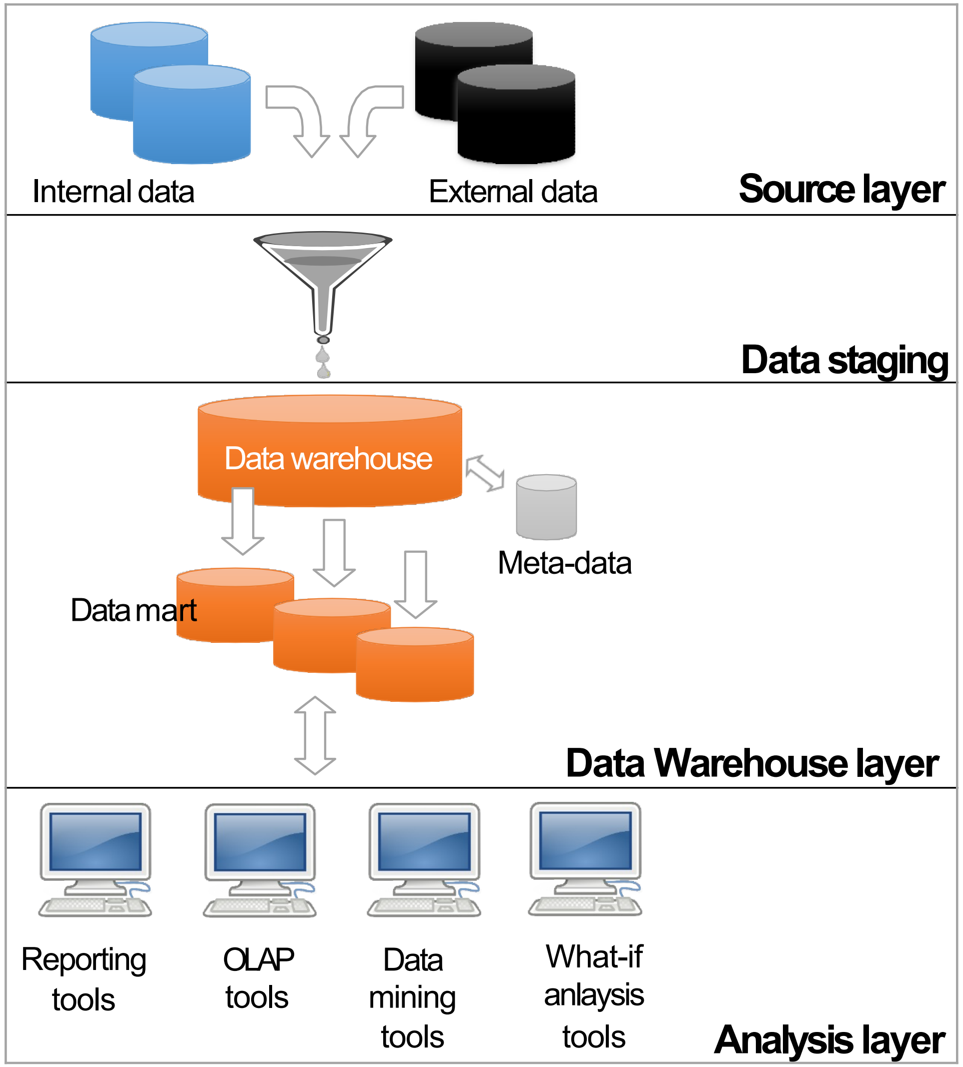
- Its goal is to minimize the amount of data stored, removing data redundancies.
- The source layer is the only layer physically available.
- DWH is implemented as a multidimensional view of operational data created by specific *middleware*.

Pros	Cons
The occupation of space is minimized.	No separation between analytical and transactional processing.



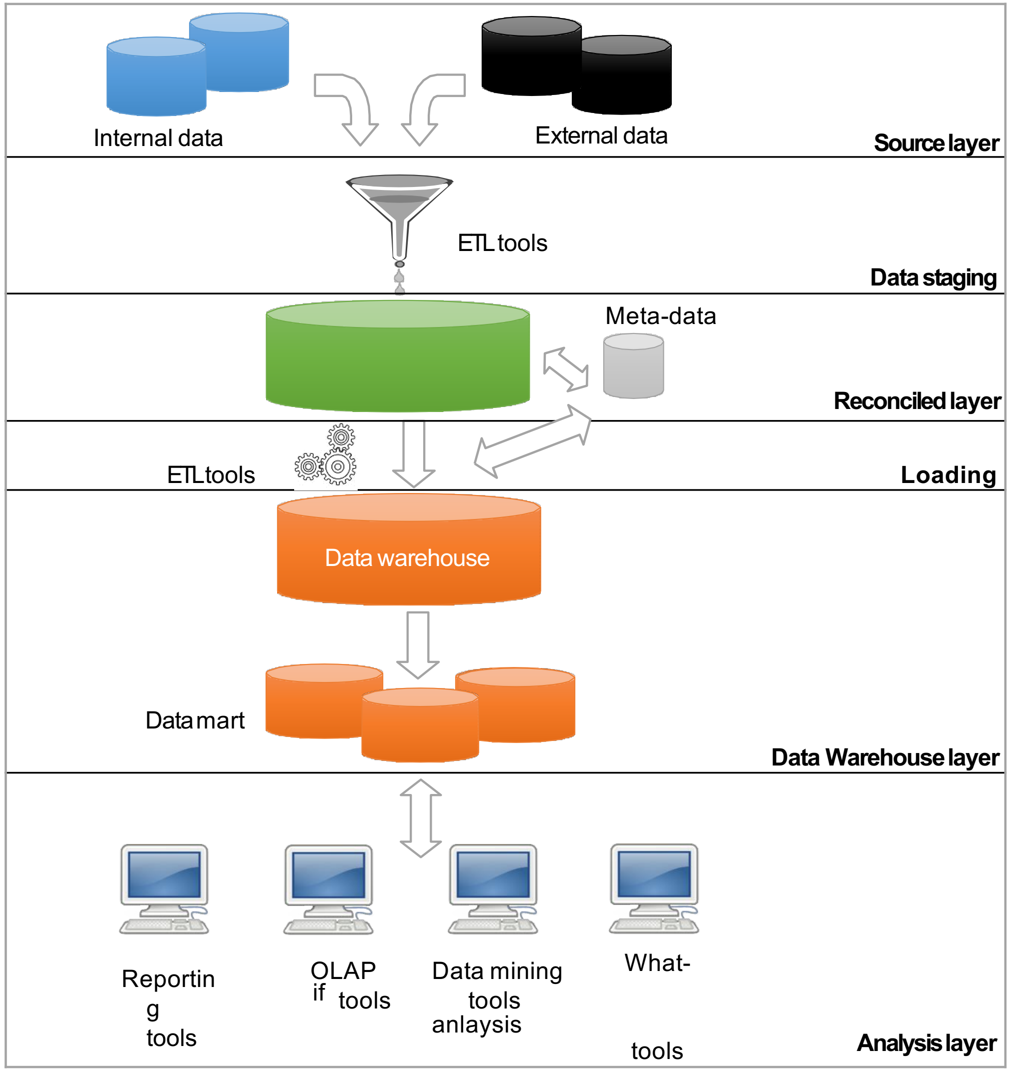
Two-Layer Architecture

- Separation between physically available sources and data warehouses.
- **Source layer**: it includes a set of heterogeneous data sources (both internal and external sources).
- **Data Staging**: the data stored to sources should be extracted, cleansed to remove inconsistencies and fill gaps, and integrated to merge heterogeneous sources into one common schema. It includes **Extraction Transformation and Loading** (ETL) procedures.
- **Data Warehouse layer**: information is stored to one logically centralized single repository that can be directly accessed or it can be used as source for creating data marts. Meta-data repositories store information on sources, data staging, data mart schemata etc.
- **Analysis layer**: it is accessed by end-user to create reports, dashboard, simulate hypothetical business scenarios etc.



Three-Layer Architecture

- **Reconciled layer**: this layer materializes operational data obtained after integrating and cleansing source data. The result data are integrated, consistent, correct, current and detailed.
- The reconciled data layer creates a common reference data model for a whole enterprise and it separates the problems of source data extraction and integration from those of data warehouse population.
- The reconciled data leads to more redundancy of operational source data.



Conceptual Modeling: The Dimensional Fact Model (DFM)

Business Intelligence Group
Università di Bologna (Prof. Stefano Rizzi, Prof. Matteo Golfarelli)

Summary

1. Conceptual design
2. Conceptual modeling
    1. DFM basic concepts
    2. DFM advanced concepts
3. Logical Design
    1. Star schema, Snowflake schema

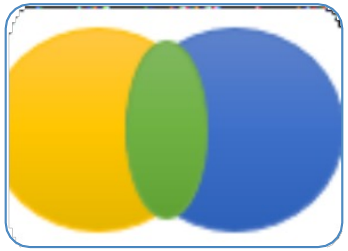
Conceptual Modeling

Requirement-driven approach

- Based on the analysis with the user to extract information on facts, measures and hierarchies etc.
- No detailed information on data sources available or sources too complex.

Data-driven approach

- The conceptual model of the data mart is based on the structure of operational sources.



Mixed approach

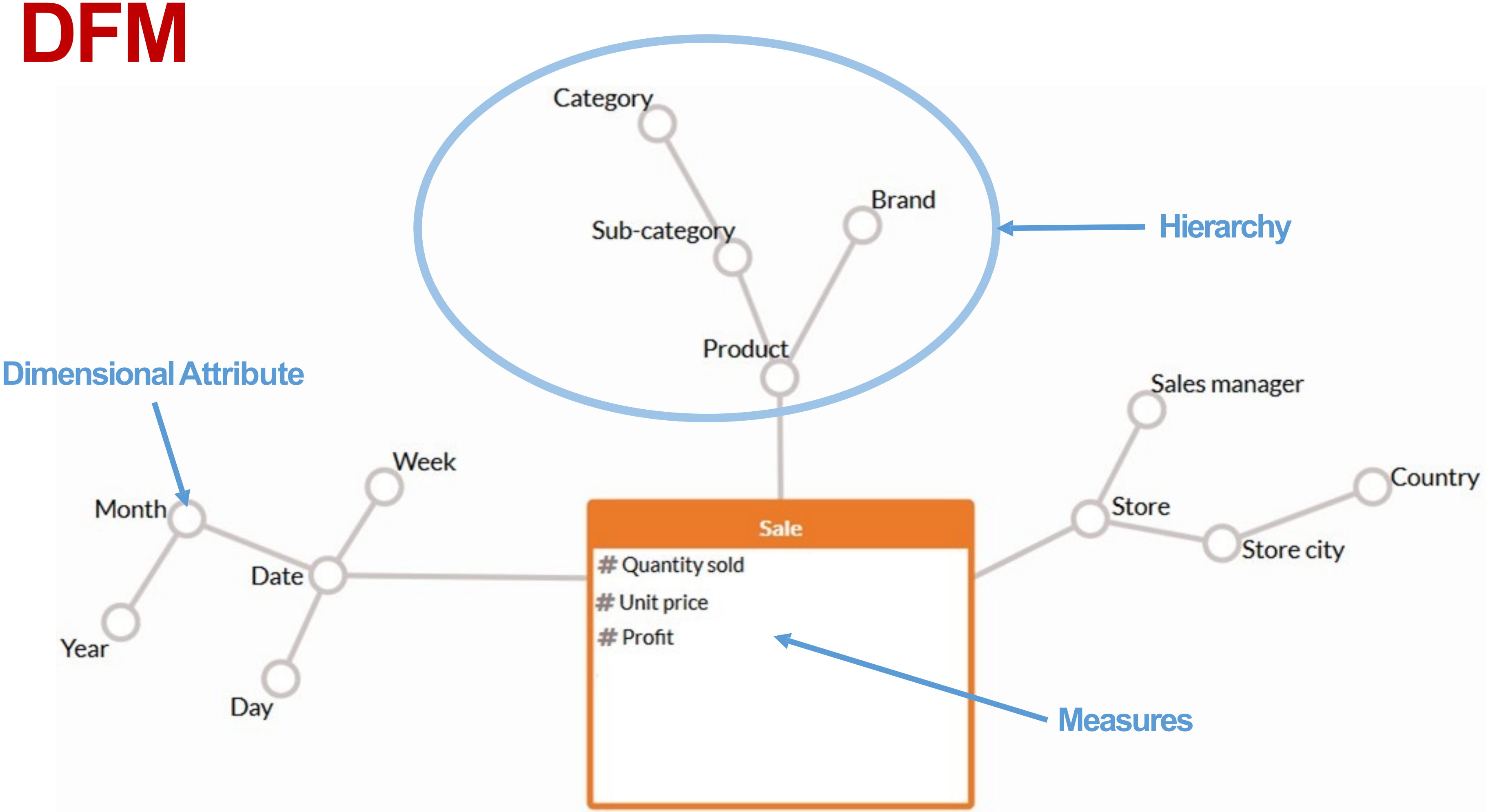
- Combination of data-driven and requirement-driven approaches.
- The result of the requirement analysis helps to refine the conceptual model derived from source analysis.

DFM

The DFM is a conceptual model created specifically to function as data mart design support. It is graphic and based on the multidimensional model.

Objectives:
Provide support to conceptual design
Create an environment in which user queries may be formulated intuitively
Favor communication between designers and end users to formalize requirement specifications
Build a stable platform for logical design
Provide clear and effective design documentation

DFM


Category
Brand
Sub-category
Hierarchy
Product
Dimensional Attribute
Sales manager
Week
Country
Month
Store
Sale
Store city
# Quantity sold
Date
# Unit price
Year
# Profit
Day
Measures

Basic concepts

Concept	Description	Example
Fact	It is a concept relevant to decision-making processes. It typically models a set of events taking place within a company	Sales, purchases, orders
Measure	It is a numerical property of a fact and describes a quantitative fact aspect that is relevant to analysis.	Quantity, revenue, discount
Dimension	It is a fact property with a finite domain and describes an analysis coordination.	Date, product, store
Dimensional attribute	Dimensions and other possible attributes, always with discrete values, that describe them.	Category of product, month
Hierarchy	It is a directed tree whose nodes are dimensional attributes and whose arcs model many-to-one associations between dimensional attribute pairs. It includes a dimension, positioned at the tree's root and all of the dimensional attributes that describe it.	Date->Month->Year

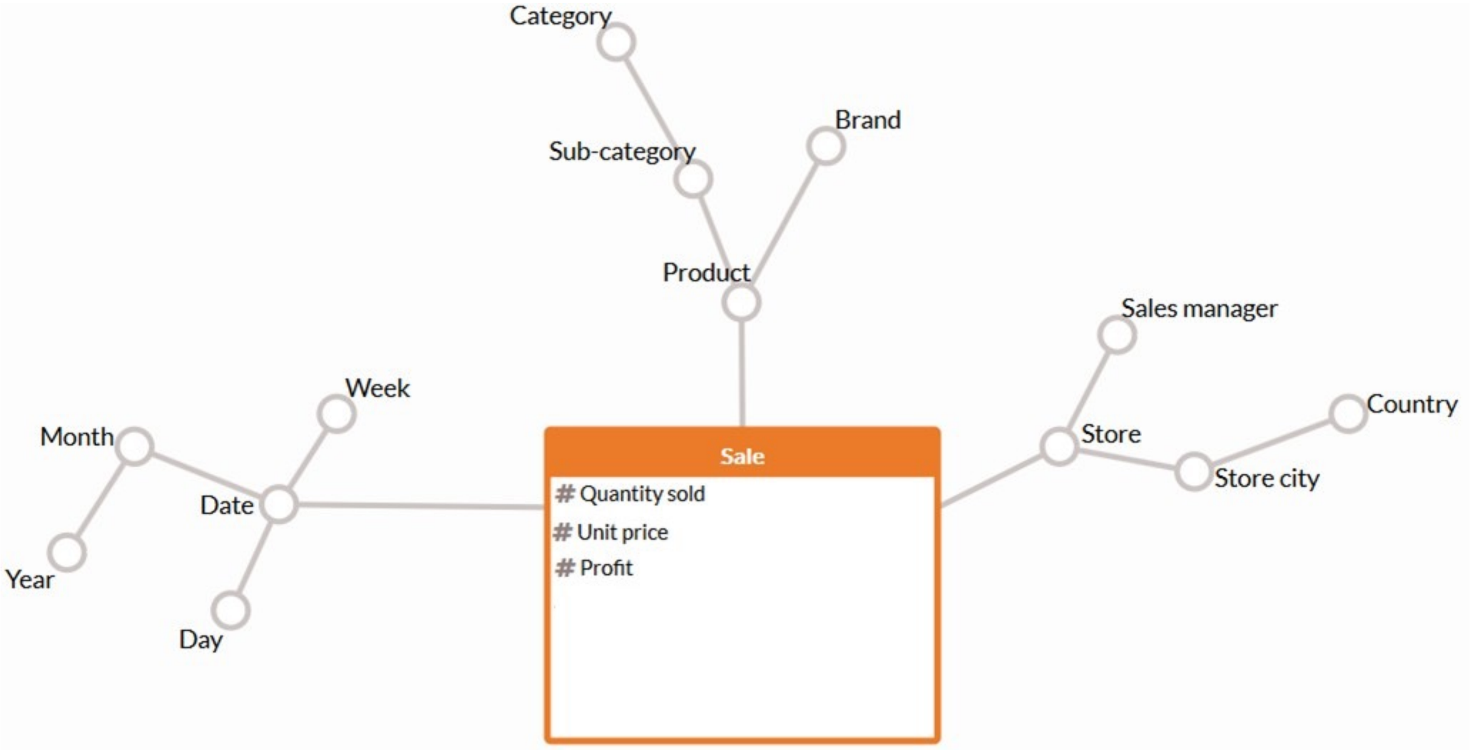
Basic concepts

Concept	Description
Primary event	It is a particular occurrence of a fact, identified by on n-ple made up of a value for each dimension. A value for each measure is associated with each primary event.
Secondary event	Given a set of dimensional attributes, each n-ple of their values identifies a secondary event that aggregates all of the corresponding primary events. Each secondary event is associated with a value for each measure that sums up all the values of the same measure in the corresponding primary events.

Examples of DMs and facts

Context	Data mart	Fact
Trade and manufacturing	Production	Orders, inventory
	Marketing	Customer fidelity, advertising
Financial services	Bank	Bank account, bank transfer
	Services	Credit card
Healthcare	Emergency services	Admissions
Telecommunications	Customer care	Customer satisfaction
	Network analysis	Data Traffic, Voice Traffic

Primary Event



Date	Store	Product	Quantity sold	Profit	Unit price
01/03/2015	Central store	Coffee	54	100	5

Secondary Event

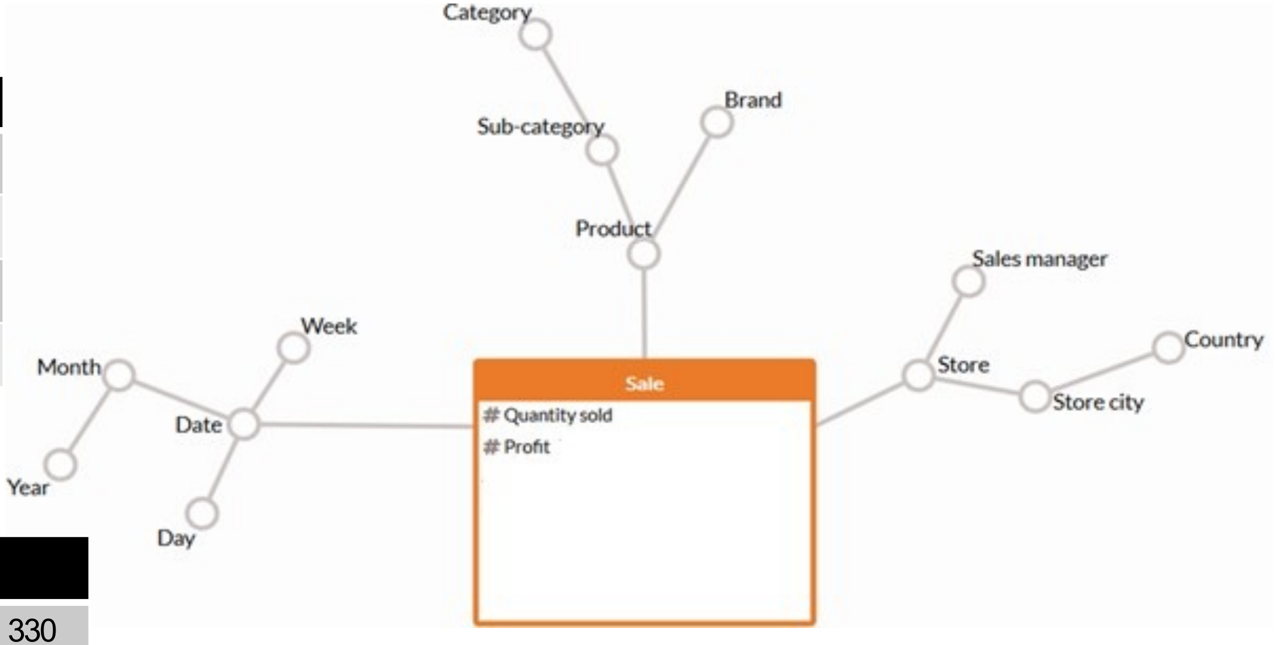
Primary events

Date	Store	Product	Qty sold	Profit
01/03/15	Central store	Milk	20	60
01/03/15	Central store	Coke	25	50
02/03/15	Central store	Bread	40	70
10/03/15	Central store	Wine	15	150

SUM SUM

Secondary event

Month	Store	Category	Qty sold	Profit
March 2015	Central store	Food and Beverages	100	330



Additivity

Aggregation requires the definition of a suitable operator to compose the measure values that mark primary events into values to be assigned to secondary events.

Measure classification:
Flow measures: refer to a timeframe, at the end of which they are evaluated cumulatively (e.g. quantity sold).
Level measures: are evaluated at particular times (e.g. number of products in inventory).
Unit measures: are evaluated at particular times but are expressed in relative terms (e.g. unit price).

Additivity

A measure is called **additive** along a dimension when you can use the SUM operator to aggregate its values along the dimension hierarchy. If this is not the case, it is called non-additive. A **non-additive** measure is **non-aggregable** when you can use no aggregation operator for it.

	Temporal Hierarchies	Non-temporal Hierarchies
Flow measures	SUM, AVG, MIN, MAX	SUM, AVG, MIN, MAX
Level measures	AVG, MIN, MAX	SUM, AVG, MIN, MAX
Unit measures	AVG, MIN, MAX	AVG, MIN, MAX

Aggregation operator classification

- **Distributive**: calculating aggregates from partial aggregates (e.g. SUM, MIN, MAX)
- **Algebraic**: requiring the usage of additional information in the form of a finite number of support measures to correctly calculate aggregates from partial aggregates (e.g. AVG)
- **Holistic**: calculating aggregates from partial aggregates only via an infinite number of support measures (e.g. RANK)

Distributive operator - example

		Quantity sold	2015		2014	
Category	Type	Product	Jan-15	Feb-15	Jan-14	Feb-14
Food and Beverages	Dairy products	White milk	90	90	60	80
		Chocolate milk	60	80	70	70
		Yogurt XY	20	30	30	35
	Beverages	Cola	20	10	35	30
		Orange Juice X	50	60	60	45

SUM

	2015		2014	
Type	gen-15	feb-15	gen-14	feb-14
Dairy products	170	200	160	185
Beverages	70	70	95	75

Secondary events – pattern {Type, Month}

SUM

Type	2015	2014
Dairy products	370	345
Beverages	140	170

Secondary events – pattern {Type, Year}

Algebraic operator - example

Unit price

			2015		2014	
Category	**Type**	**Product**	**Jan-15**	**Feb-15**	**Jan-14**	**Feb-14**
Food and Beverages	Dairy products	White milk	2	2	2,2	2,5
		Chocolate milk	1,5	1,5	2	2,5
		Yogurt XY	1,75	3	3	3
	Beverages	Cola	1	1,2	1,5	1,5
		Orange Juice X	1,5	1,5	2	1,5

AVG

Secondary events – pattern {Type, Month}

		2015		2014	
Category	**Type**	**Jan-15**	**Feb-15**	**Jan-14**	**Feb-14**
Food and Beverages	Dairy products	1,75	2,17	2,40	2,67
	Beverages	1,25	1,35	1,75	1,50

✗ AVG

Secondary events – pattern {Category, Month}

	2015		2014	
Category	**Jan-15**	**Feb-15**	**Jan-14**	**Feb-14**
Food and Beverages	1,50	1,76	2,08	2,09

AVG

Secondary events – pattern {Category, Month}

	2015		2014	
Category	**Jan-15**	**Feb-15**	**Jan-14**	**Feb-14**
Food and Beverages	1,55	1,84	2,14	2,02

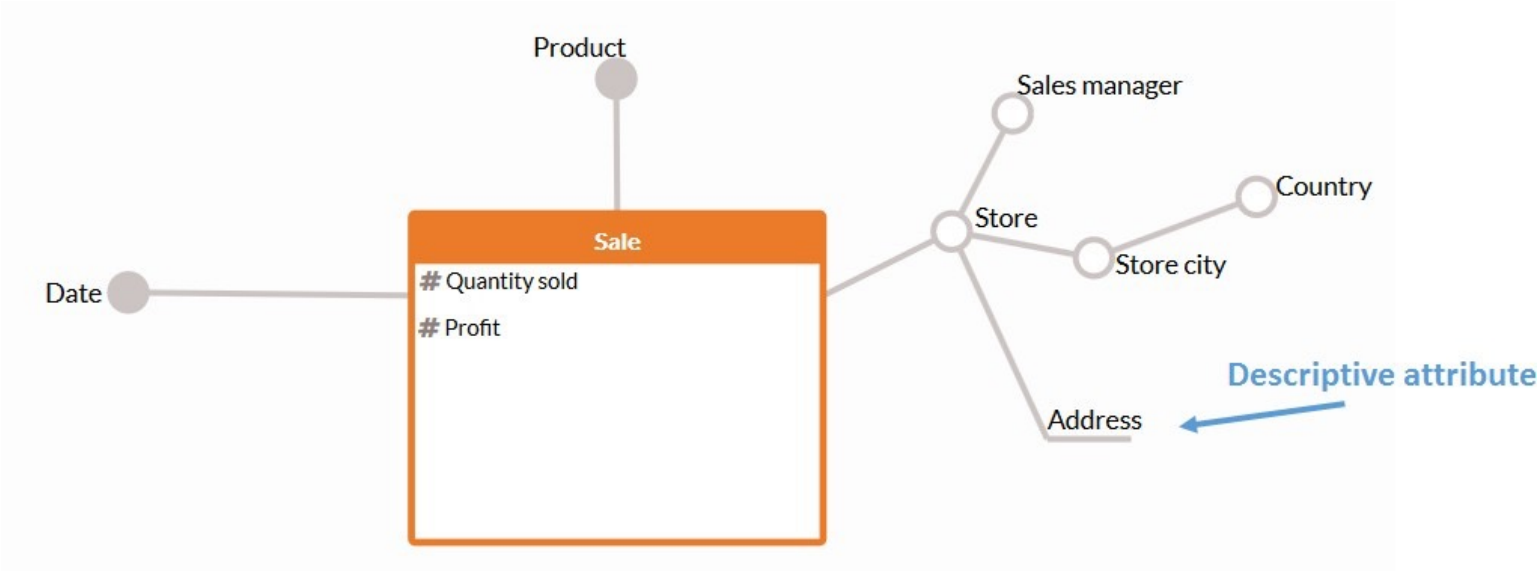
		2015		2014	
Category	**Type**	**Jan-15**	**Feb-15**	**Jan-14**	**Feb-14**
Food and Beverages	Dairy products	3	3	3	3
	Beverages	2	2	2	2

We need to use the support measure **COUNT** that counts the number of primary events that make up each secondary event.

DFM – advanced concepts

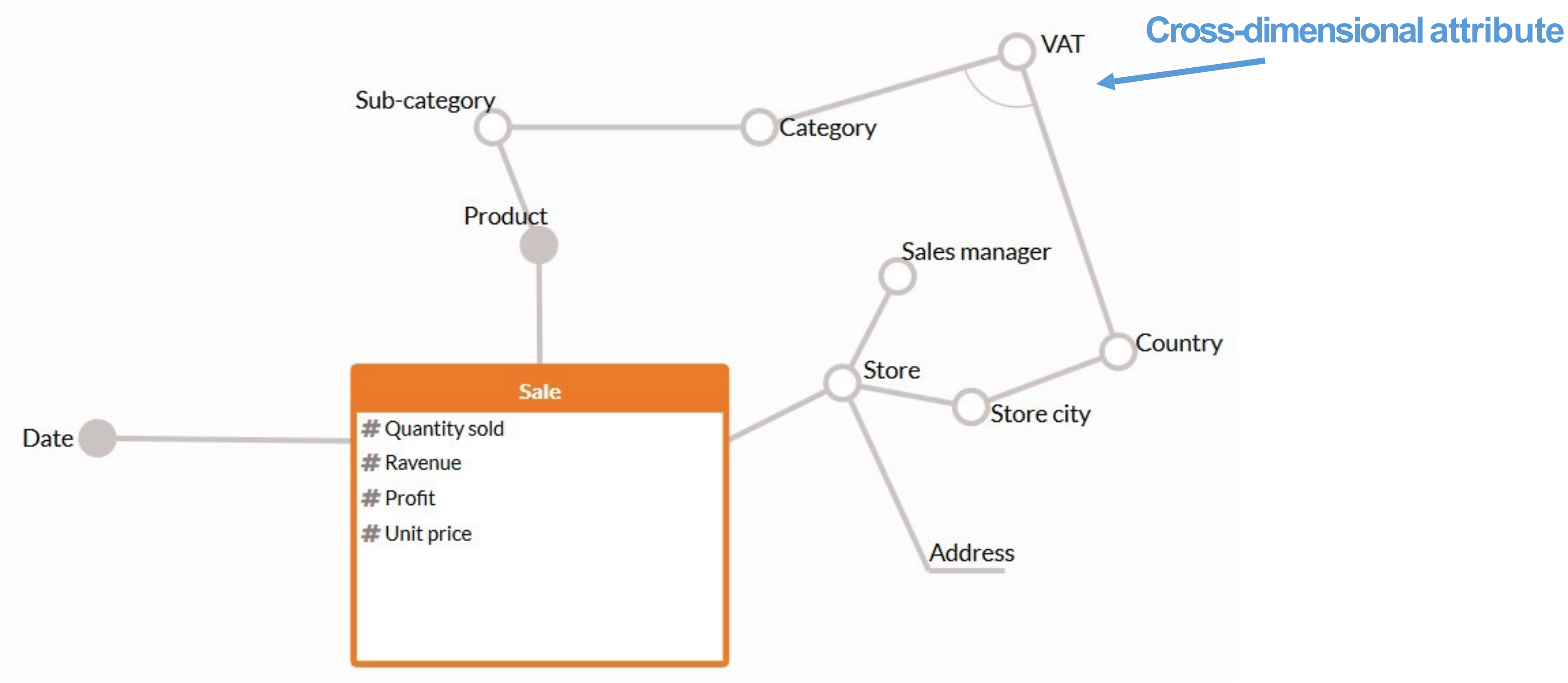
- Descriptive attributes
- Cross-dimensional attributes
- Convergence
- Shared Hierarchies
- Multiple arcs
- Optional arcs
- Incomplete hierarchies
- Recursive hierarchies

Descriptive attributes



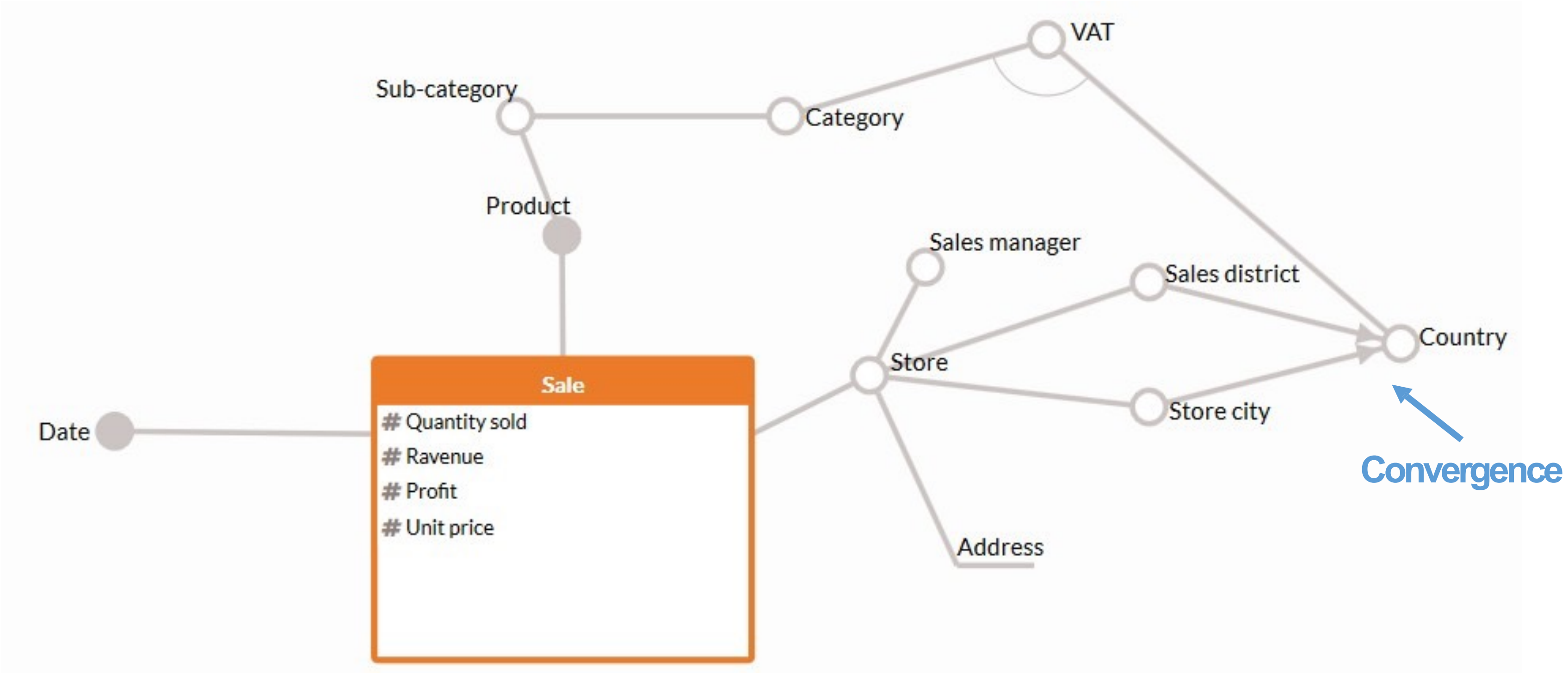
Adescriptive attribute is used to give additional information to a specific dimensional attributes, but it is not used as aggregation criteria.

Cross-dimensional attributes



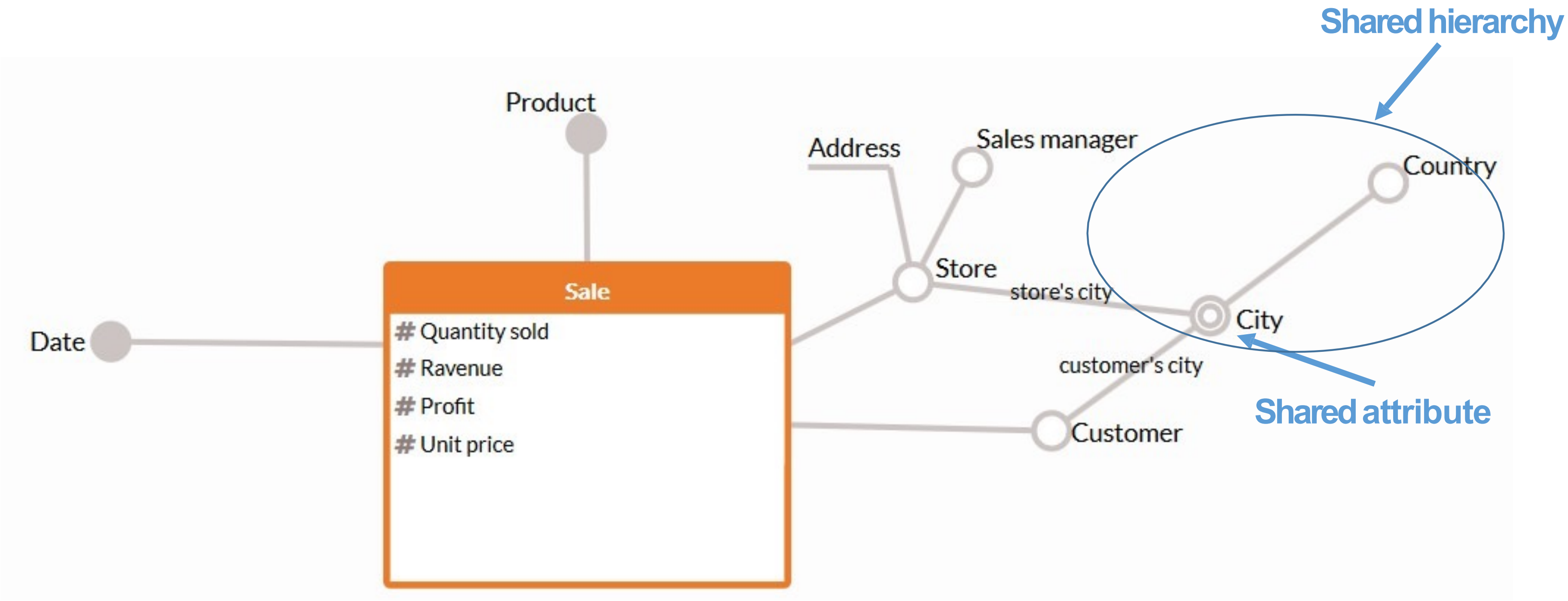
It is a dimensional or descriptive attribute whose value is defined by the combination of two or more dimensional attributes, possibly belonging to different hierarchies.

Convergence



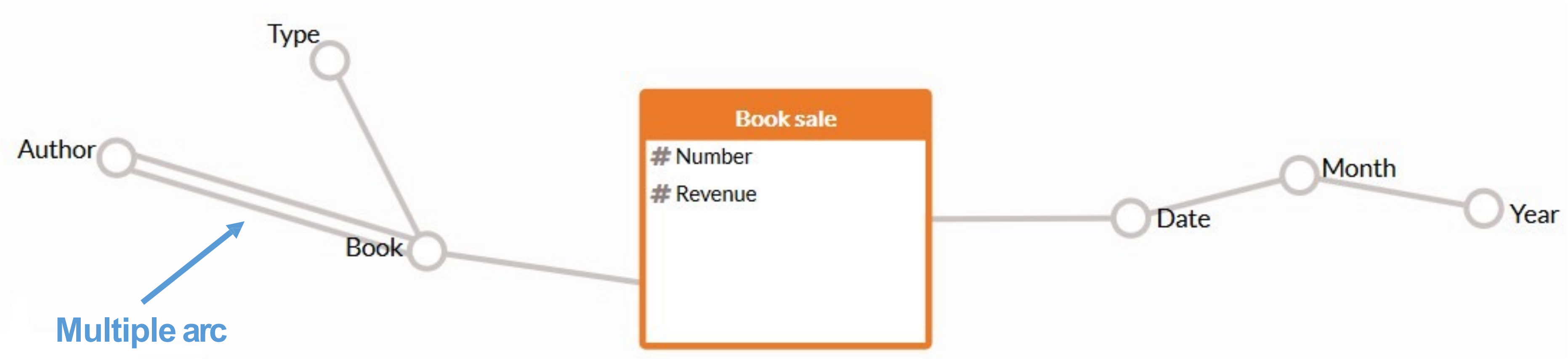
Two or more arcs belonging to the same hierarchy and ending at the same dimensional attribute.

Shared hierarchies



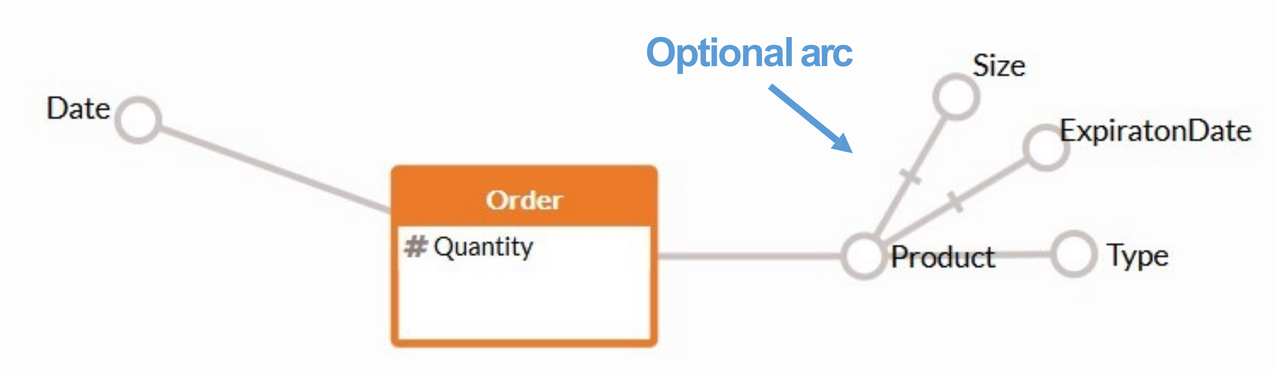
Adouble circle represents and emphasizes the first attribute to be shared (e.g., City). All descendants of the shared attribute are shared, too. For each incoming arc a role must be added (e.g., customer’s city).

Multiple arcs



The meaning of a multiple arc that goes from an attribute called **a** (e.g., book) to an attribute called **b** (e.g., author) is that a many-to-many association exists between **a** and **b**.

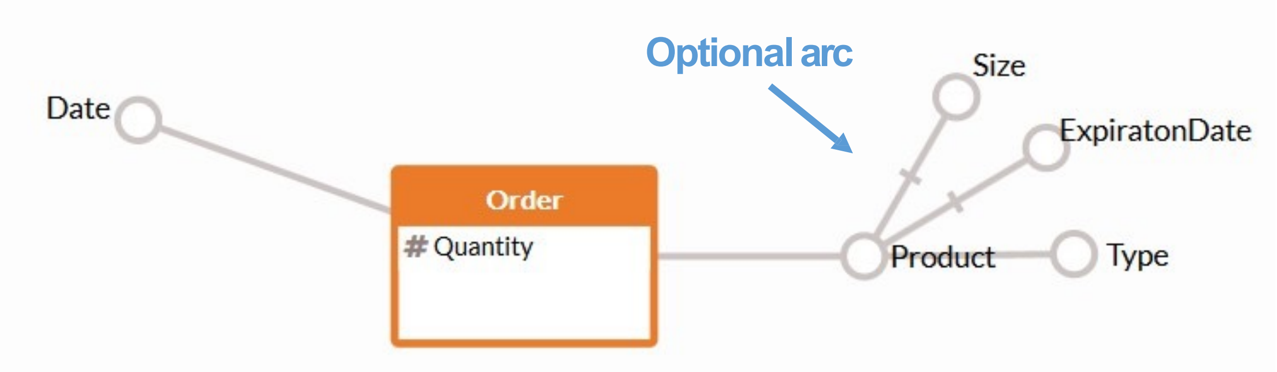
Optional arcs



Optionality is used to model scenarios for which an association represented in a fact schema is not defined for a subset of events.

Example: for one or more value of *Product*, the *ExpirationDate* and all the possible descendants in the hierarchy may be undefined.

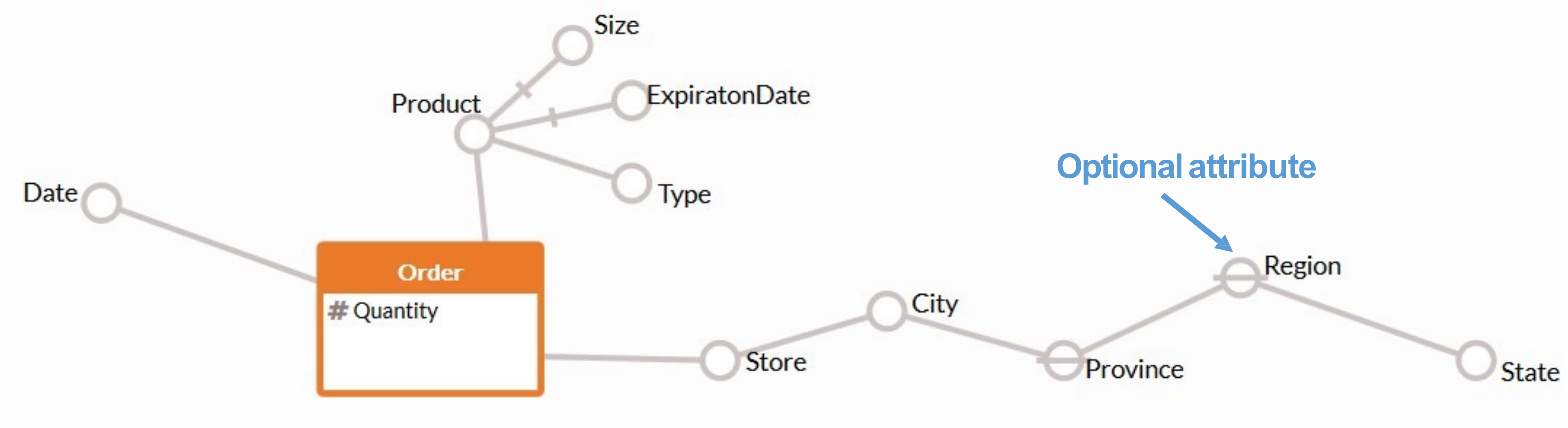
Optional arcs - Coverage



When two or more optional arcs exit from the same attribute ***a*** (e.g., Product), you can specify their coverage:

- The coverage is **total** if the value of at least one of the children is linked to each value of **a**. If, instead, values of **a** exist for which all of the children are undefined, the coverage is **partial**.
- The coverage is **disjoint** if you have a value for at most one of the children corresponding to each value of **a**. If, instead, values of **a** exist linking to values of two or more children, the coverage is **overlapping**.
- Total or Partial? Disjoint or overlapping?

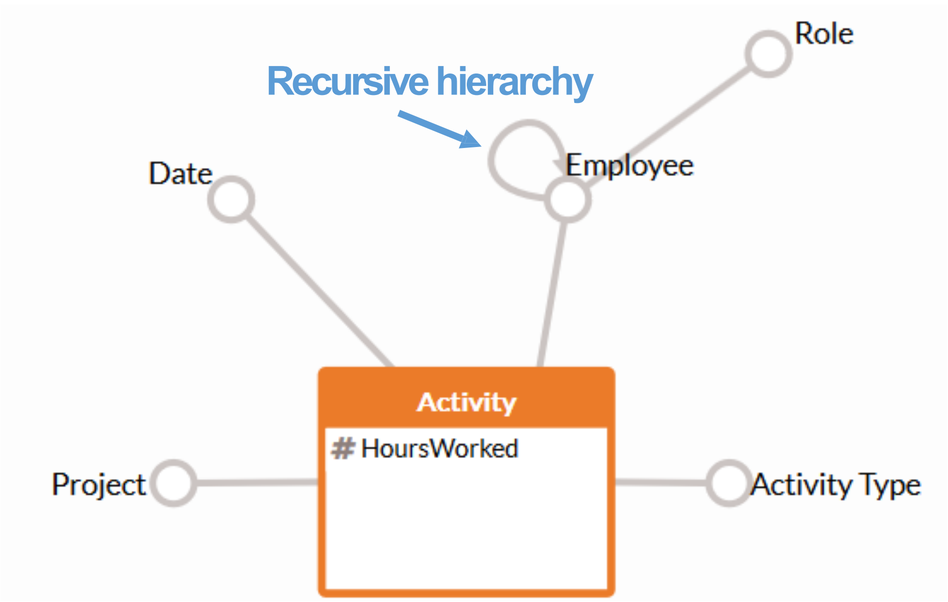
Incomplete hierarchies



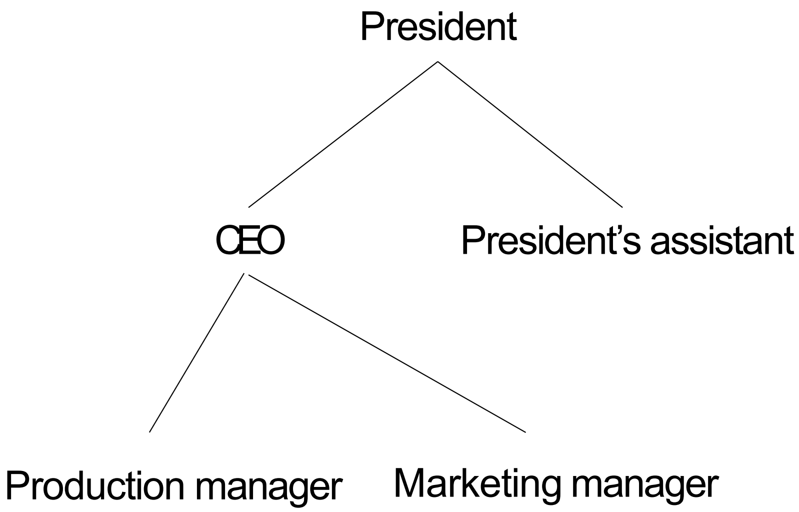
Optional attribute

An incomplete hierarchy is one in which one or more levels of aggregation prove missing in some instances (because they are not known or defined).

Recursive hierarchies



Example: role hierarchy in an unbalanced company organization chart


President

CEO

President’s assistant

Production manager

Marketing manager

A recursive hierarchy represents a parent-child relationship among the levels.

Logical Design

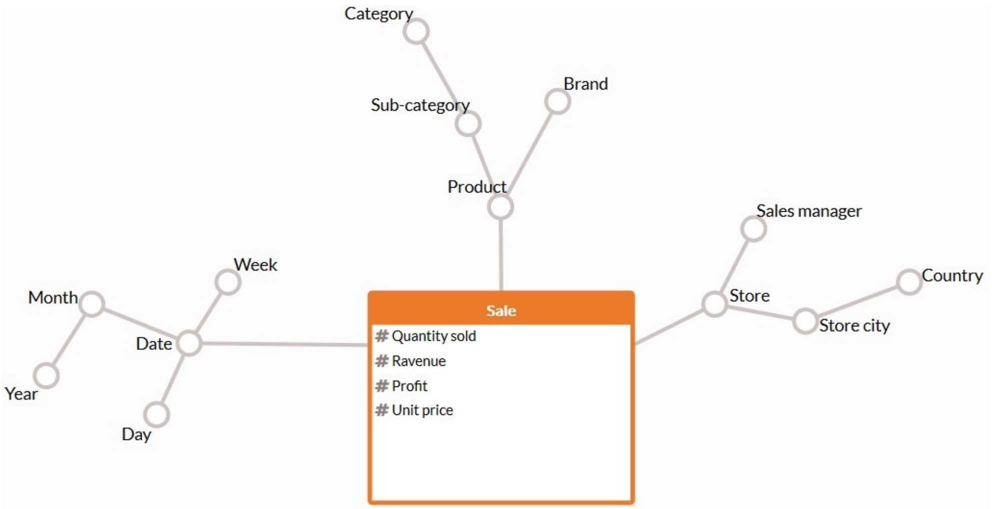
The logical design phase defines the data structures (e.g., set of tables and relationships between tables) that represent data marts according to the preselected logical model and optimizes performance by fine-tuning these structures.

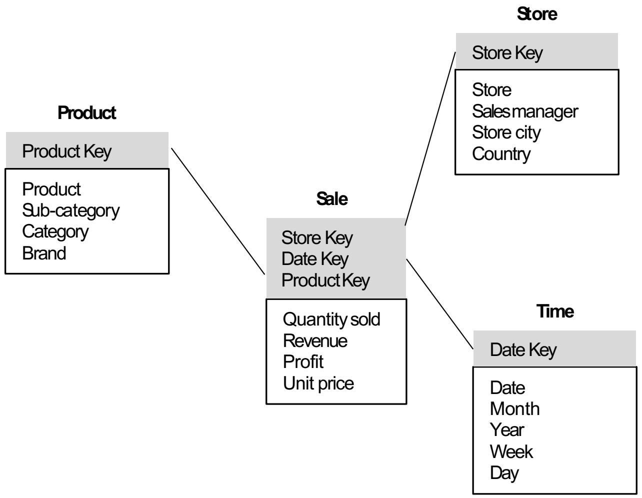
The logical design phase includes a set of steps that, starting from the conceptual schema, make it possible to define the logical schema of a data mart. Three main steps to implement a logical schema in a relational DBMS:

- Translating fact schemata into logical schemata: mainly star or snowflake schemata.
- Materializing views: set of secondary view that aggregate primary view data to improve query performance.
- Fragmenting fact tables vertically and horizontally.

Star schema

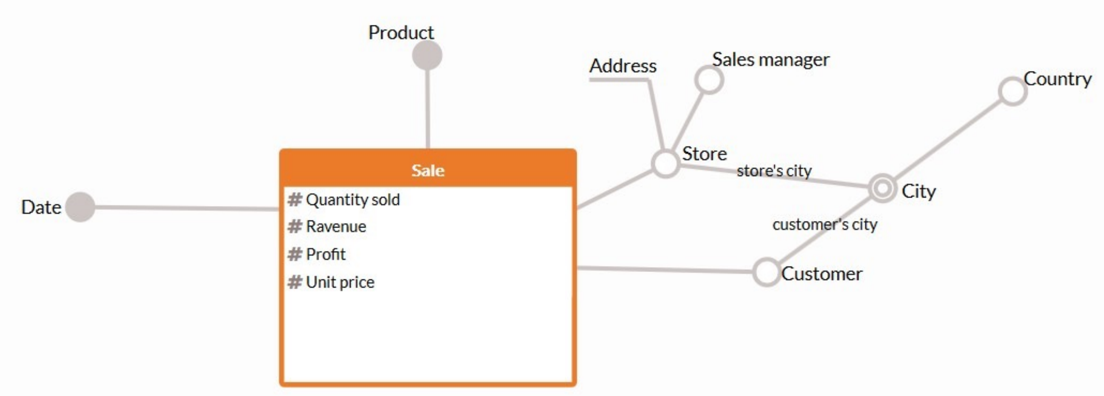
A star schema is characterized by fact tables and dimension tables. A fact table contains all the measures and descriptive attributes linked to a fact. A dimensional table is created for each dimension and it includes all the hierarchy attributes.





Snowflake schema

Astar schema variant with dimension tables partially normalized.



Product

Address

Sales manager

Country

Sale

# Quantity sold

# Ravenue

# Profit

# Unit price

Store

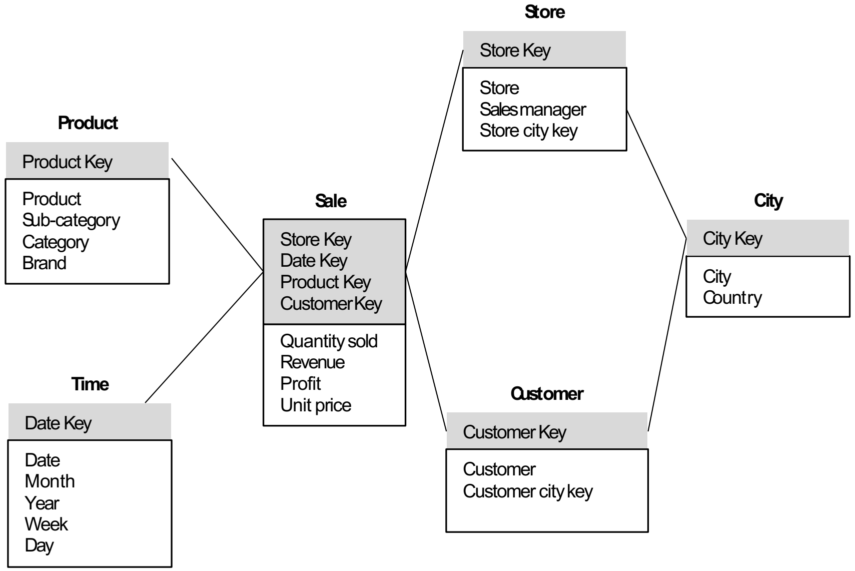
store's city

City

customer's city

Date

Customer



Store

Store Key

Store
Sales manager
Store city key

Product

Product Key

Product
Sub-category
Category
Brand

Sale

Store Key
Date Key
Product Key
Customer Key

Quantity sold
Revenue
Profit
Unit price

City

City Key

City
Country

Time

Date Key

Date
Month
Year
Week
Day

Customer

Customer Key

Customer
Customer city key

Exercise on conceptual modeling

A telecommunication provider wants to create a data mart to monitor its network performance (both Data and Voice traffic) on the European area. Data come from different countries in Europe (and from its main cities): Italy, Spain, Albania, Germany, Malta, France, Portugal and UK. These countries are divided into different groups:

- Main group: Italy, France, Spain, UK.
- Small group: Albania, Germany, Malta, Portugal.

Data are analyzed on daily/weekly/monthly-basis. Moreover the provider wants to analyze the information depending on the type of technology used (e.g., 2G,3G).